

Amendments to the Claims:

This listing of claims will replace all prior versions, and listings of claims in the application:

Listing of Claims:

- 1 1. (Currently amended): A method for identifying protocol encapsulation in
2 received network data comprising providing language definition including a grammar; receiving
3 incoming network data and processing it-said incoming network data in accordance with a formal
4 language processing technique using said language definition, said processing including parsing
5 said network data using said grammar, said network data being organized into data packets.
- 1 2. (Original): The method of claim 1 wherein said grammar is a grammar
2 graph, the method further including providing a deterministic finite automaton (DFA)
3 representing said grammar graph.
- 1 3. (Original): The method of claim 1 further including scanning said
2 incoming network data using lexical token scanning to produce plural lexical tokens, said step of
3 parsing including parsing said lexical tokens.
- 1 4. (Original): The method of claim 3 wherein said lexical scanning includes
2 providing a set of regular expressions.
- 1 5. (Original): The method of claim 3 further including providing a
2 deterministic finite automaton (DFA), said DFA including a representation of said lexical tokens
3 and said grammar, said step of scanning including recognizing lexical tokens contained in said
4 data packets using said DFA, said step of parsing including identifying grammatical structure
5 among said lexical tokens using said DFA to identify protocol encapsulation in said incoming
6 network data.

1 6. (Currently amended): In a data packet network switching device, a
2 method for processing data packets comprising:
3 providing a language definition including a grammar;
4 receiving plural data packets, each having a length not necessarily equal to one
5 another; and
6 for each data packet, processing ~~it~~ said data packet according to a formal language
7 processing technique using said language definition including lexically scanning said data packet
8 to produce plural lexical tokens, parsing said lexical tokens using said grammar to produce one
9 or more identified protocols, and processing said data packet based on said identified protocols.

1 7. (Original): The method of claim 6 further including compiling said
2 grammar to produce a grammar graph.

1 8. (Original): The method of claim 7 wherein said lexical scanning includes
2 providing regular expressions for identifying said lexical tokens.

1 9. (Original): The method of claim 8 further including compiling said
2 regular expressions are into a deterministic finite automaton (DFA).

1 10. (Original): The method of claim 9 further including incorporating said
2 grammar graph into said DFA.

1 11. (Previously presented): In a data packet receiving and forwarding device,
2 a method for processing data packets comprising a stream of data, said method comprising:
3 receiving a description of grammar rules in the form of a grammar packet
4 classification language;
5 compiling said grammar packet classification language to produce a grammar
6 graph;
7 configuring a programmable grammatical packet classifier with said grammar
8 graph;

9 processing said data stream in accordance with a formal language processing
10 technique using said grammar packet classification language including parsing said data stream
11 with said grammatical packet classifier to identify a protocol structure in a received data packet;
12 and
13 processing said received data packet in accordance with said protocol structure.

1 12. (Original): The method of claim 11 further including:
2 receiving a description of classification rules in a lexical classification language;
3 compiling said classification language to produce a deterministic finite automaton
4 (DFA) comprising plural states;
5 configuring said hardware packet classifier with said DFA; and
6 scanning said data stream with said hardware packet classifier to produce plural
7 lexical tokens,
8 wherein said parsing is a step of parsing said lexical tokens.

1 13. (Original): The method of claim 12 wherein said grammar graph is
2 incorporated into said DFA.

1 14. (Original): The method of claim 12 wherein said lexical classification
2 language includes regular expressions.

1 15. (Original): The method of claim 14 wherein said regular expressions
2 include arithmetic and logic operations.

1 16. (Original): The method of claim 15 wherein said regular expressions
2 further include skip operations.

1 17. (Original): The method of claim 16 wherein said regular expressions
2 further include data storage operations.

1 18. (Previously presented): A network data packet classifier comprising:
2 an input port for receiving network data packets comprising a stream of data;

3 a memory assemblage configured with data representing a deterministic finite
4 automaton (DFA), said DFA defined by a language definition and representing a grammar graph
5 and plural regular expressions; and
6 decompression logic operatively coupled to said memory assemblage and
7 configured to process said stream of data according to a formal language processing technique
8 using said language definition including a step to scan said stream of data with said DFA to find
9 a matching one of said regular expressions thereby producing plural lexical tokens,
10 said decompression logic further configured to parse said lexical tokens with said
11 DFA to identify a protocol structure in a received network data packet,
12 wherein processing of said network data packet depends on said protocol
13 structure.

1 19. (Original): The classifier of claim 18 wherein some of said regular
2 expressions include arithmetic instructions and logic instructions, said memory assemblage
3 further configured to contain said instructions, the classifier further including an arithmetic logic
4 unit operatively coupled to said decompression logic and configured to execute said instructions.

1 20. (Original): The classifier of claim 19 further including at least one register
2 operatively coupled to said arithmetic logic unit, said arithmetic logic unit further configured to
3 store data into said register in response to a save instruction.

1 21. (Original): The classifier of claim 19 further including skip logic
2 operatively coupled to said logic component and configured to skip over an amount of data in
3 response a skip instruction.

1 22. (Original): The classifier of claim 18 wherein said network data packets
2 can vary from one packet to another.

1 23. (Original): The classifier of claim 18 wherein said DFA is in compressed
2 form.

1 24. (Original): The classifier of claim 23 wherein said DFA comprises plural
2 non-default states and plural default states, and said memory assemblage comprises a base
3 memory, a next-state memory, and a default-state memory; said base memory configured to
4 contain address locations of said next-state memory, said next-state memory representing all of
5 said non-default states, said default-state memory representing all of said default states.

1 25. (Original): The classifier of claim 24 wherein said memories are random
2 access memories.

1 26. (Original): The classifier of claim 24 wherein said memories are read-
2 only memories.

1 27. (Previously presented): A network packet classifier comprising:
2 means for receiving an incoming network packet; and
3 means for identifying protocol structure in said network packet including means
4 for processing said network packet in accordance with a formal language processing technique
5 using a language definition, including a step of scanning to match patterns in its constituent data
6 against plural regular expressions to produce lexical tokens and means for parsing through said
7 lexical tokens using a grammar, said regular expressions and said grammar being defined by said
8 language definition.

1 28. (Original): The classifier of claim 27 wherein said means for scanning
2 includes a memory component configured with data to represent a deterministic finite automaton
3 (DFA).

1 29. (Original): The classifier of claim 28 wherein said memory component is
2 further configured to include said grammar.

1 30. (Original): The classifier of claim 27 wherein said regular expressions
2 include arithmetic specifiers and said means for classifying includes an arithmetic logic unit
3 configured to perform operations in accordance with said arithmetic specifiers.